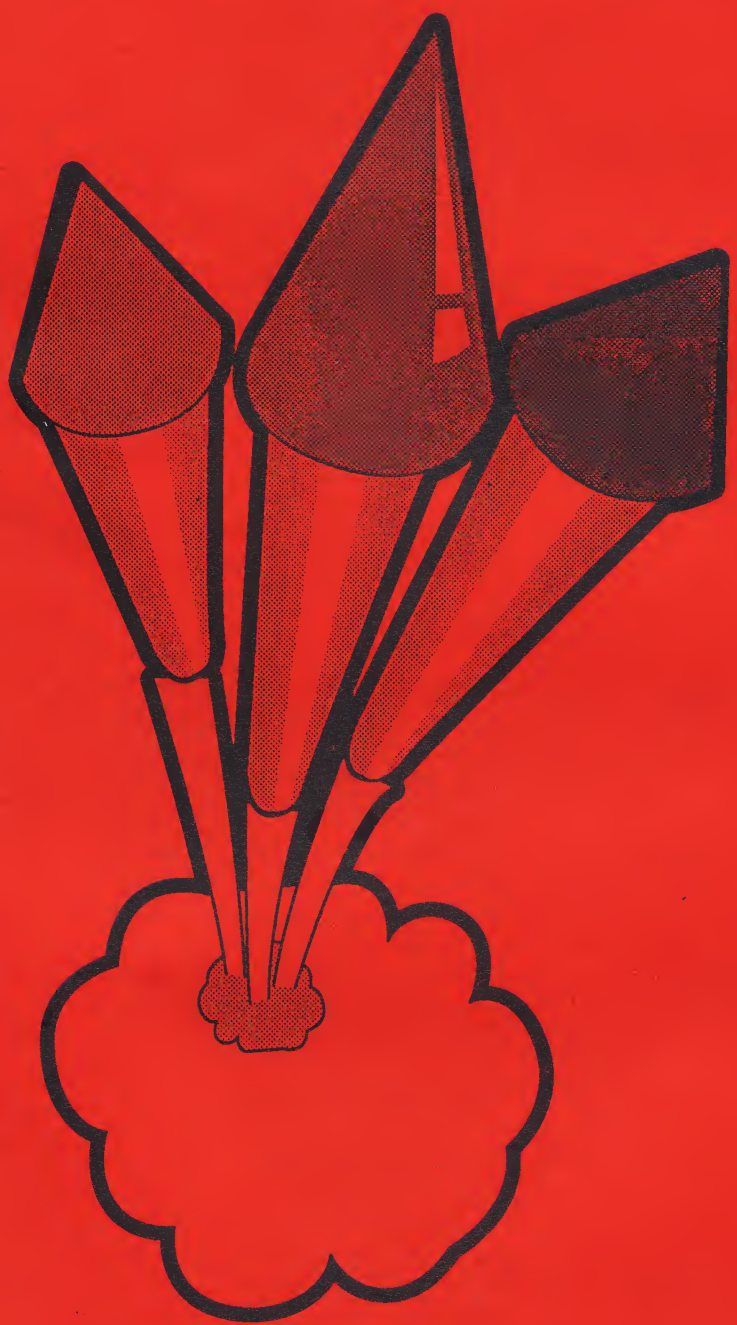
South • Australian • Apple • Users • Club

APPLESAUCE



Guy Fawkes - for those who remember!

November 1990 - $2.50

Registered by Australia Post Publication No. SBH0229

South Australian Apple Users Club, Inc.

Postal address
SAAUC, Inc.
PO Box 322
Prospect SA 5082

Club Committee 1990-1991

President
Bob Owen (Apple][GS & Mac)
Ph (08) 293 1313

Vice President
Peter Jenkins (Mac)
Ph (08) 2435692

Treasurer
Peter McKay (Apple][GS)
Ph (08) 262 3387

Secretary
Ian Bagust (Mac II)
Ph (08) 293 7183

Newsletter Editor
Bill Wardrop (Mac Plus)
Ph(08) 251 2154

Apple][Editor
Keith Zerna (Apple][GS)
Ph (08) 2650364

Program Co-Ordinator
Peter Jenkins (Mac)
Grant Ward (Apple][)

Macintosh Disk Librarian
Brenton Smith (Mac SE)

Apple][/GS Disk Librarian
David Pettinger (Apple //c)

Magazine Librarian - Apple][
George Wissenburg (Apple][GS)

Magazine Librarian - Mac
Michael Coleman (Mac)

Membership Fees

Individuals or Schools $30.00
Corporations / Businesses $50.00
Extended memberships available

Library Disks

Public Domain disks
Apple][(5.25") $4.00 each
Apple][GS (3.5") $7.00 each
Macintosh $7.00 each
(10 or more $5.00 each)

Boxes of 10 blank disks
3.5" DSDD $23.00
3.5" DSHD $50.00
Disk Catalogues 1990 $2.00 each

Disclaimer

The SAAUC while making every endeavour to ensure Public Domain disks are free of any virus or other fault, no responsibility is accepted for any damage to computer equipment or loss of data which may be caused by these disks or programs. All programs supplied on club disks have been written using original Apple equipment; no guarantee is given by the club for use of programs on Apple compatible computers.

Postal rates for disks

1 to 9 disks add $2.00
10 disks add $3.00
Greater than 10 disks add $4.00
The above rates include Australia Post insurance worth $50.
Please add an additional $2.50 for insurance cover worth $200.

Public domain disk purchases of $20 or more from S.A. country members (Postcodes 5200 or greater) or interstate members will be posted FREE. The additional insurance charge of $2.50 is still payable if you require the extra cover.

Information page

Meetings

Are held on the 1st and 3rd Fridays of each month at the rear of the Prospect Town Hall, Prospect Road, Prospect.

Contributing Articles

Articles will be accepted from anyone wishing to contribute material for publication in the newsletter, provided it is oriented towards Apple computers. It can be of any topic relating to Apple computers, and may be typed printed normally spaced on standard A4 paper, or preferably on disk as a plain text document. Send articles to the Newsletter Editor c/- above postal address or hand to a club committee member at any meetings. All disks will be returned.

Members who contribute an article with a length of two pages or greater will receive their choice of club disk from the library on request.

Advertising

Small advertisements (quarter page or less) from financial club members will be accepted and published free of charge provided they are signedbythe person(s) placing the advertisement.

Business advertising rates are by negotiation or as follows:
1/4 page $18.00
1/2 page $30.00
Full page $50.00
Back cover $70.00

All advertisements will be printed under the proviso that they do not conflict with the best interests of the Club.

Copyright

Applesauce is the official newsletter magazine of the SAAUC, Inc. This publication is copyright ©1989 by SAAUC, Inc. Authors and advertisers express their own opinions and are responsible for the accuracyoftheir submitted material.

Permission is granted for the reproduction of original articles contained in this issue by any non-profit organisation provided the author, title and publication credits are given. ■

Editorial

Bill Wardrop

I mentioned in last month's editorial that a new MNS disk had been received but had been badly bent in the post. With some perseverance, Bob Owen managed to straighten it enough to make a copy from it. Some articles from that disk are in this issue.

You will notice that locally written material is in short supply in this issue, so I guess I am appealing for you to take to the keyboard and put together an article, however short, on some aspect of your computing experiences.

CONTENTS

1 Cover
2 Information page
3 Editorial
4 Presidential Pen
5 Diary Dates
6 Kansas Fest Report #1
10 Ducks
11 Rajiv Mehta's speech
15 Windows 3 - No threat to Mac
16 Typestyler Review
17 Inside the MacIntosh.
19 Games #21
19 Font Story
21 The Written Word #2
23 Eleven Misteaks that .....

Note

New Public Domain Disk Prices

Apple][$4.00 each
Apple][GS $7.00 each
Macintosh $ 7.00 each

See 'Disk Catalogue 1990' for further details

To Swap:

Apple][Instant Pascal; full documentation, original software plus extra programs disk & extra reference manual: Odesta "How about a nice game of chess?" for a recent version of Appleworks with full documentation.
Jason, 272 4324

For Sale:

The following programs, all original copies with boxes and manuals.

Bruce Lee
The Bards Tale II
Hacker II
Alcazar - The Forgotten Fortress
Captain Goodnight and the Islands of Fear
Airheart (IIe, IIc, IIgs only)

All $20 each or $100 the lot

97 Blank 5.25 Disks (some still unopened) at .50c each or with above software $140. Will split

On Balance accounting package for IIe IIc, IIgs (original box, software etc) = $100

Phone (08) 2650364 and ask for Keith Zerna or see me at next meeting.

Presidential Pen

N O V E M B E R 1 9 9 0

Another year is drawing to a close. Two meetings this month and we are left with the our last meeting for the year on Dec 7th. Why raise this, you ask! Well, I do it to make you all aware that we still have some objectives to achieve this year, and there is very little time left to meet these objectives.

The special interest group nominations have now closed, those who have left their name on one or more of the lists will be contacted very shortly by the facilitator of the Speccial Interest Group.

The first meeting of the group will, hopefully, take place at the December meeting, where we have a general meeting for all members. The Mac meeting would otherwise be too close to christmas for most members.

Once again the committee will be arranging for a few nibbles and drinks for Dec 7th, so make sure you mark this date on your calender.

I missed the Committee meeting last month due to holidays in Victoria, I noticed - I am not too sure that anyone else did!!

This month promises to have a special interest with the first demonstration of the Apple][Gs Video Overlay Card. This card has been loaned to the club by Apple Australia. The first demo will be very short, as we are still feeling our way with the possibilities that this card offers. At the December meeting we will have another demo, in what will be a series of demonstrations using this card. It is my intention, providing the committee agree, that we will loan the card to members who have an interest in purchasing a card for their own use. But you must be prepared to provide either a demonstration, or write an article for Applesauce, on what you have used it for.

The Tape/Disk Training packs are proving to be very popular. These are available at no cost to the members (providing they return them within the month - otherwise you donate your deposit to the club). We are hoping to acquire further subjects to expand our Library of these training packs. We ask you to copy at least the disk, and work with **your** disk. Whether you copy the tape is up to you. This is probably the best thing to do if you wish to use the program over a period exceeding one month. Remember that you may only copy this for your own use, we do not want members to be the source of distribution and copyright breach.

Regards
Bob Owen.

Diary Dates

General Meeting (Apple II, GS and Mac)
Friday 2nd November, 1990 7:30 p.m.

- Introduction to the Video Overlay Card for the Apple][Gs - David Pettinger.
- The Intel 80536 Graphics Co-processor - show and tell, by Graham Dicker.
- Question and Answer section
- Boxes of 3.5" disks for sale
- All new Apple][, GS and Macintosh public domain disks for sale
- Free tea, coffee and biscuits
- All are welcome - Mac users too!

Committee Members Only
Meeting will be at the airport location on Friday 9th November 1990 at 7:30 p.m.

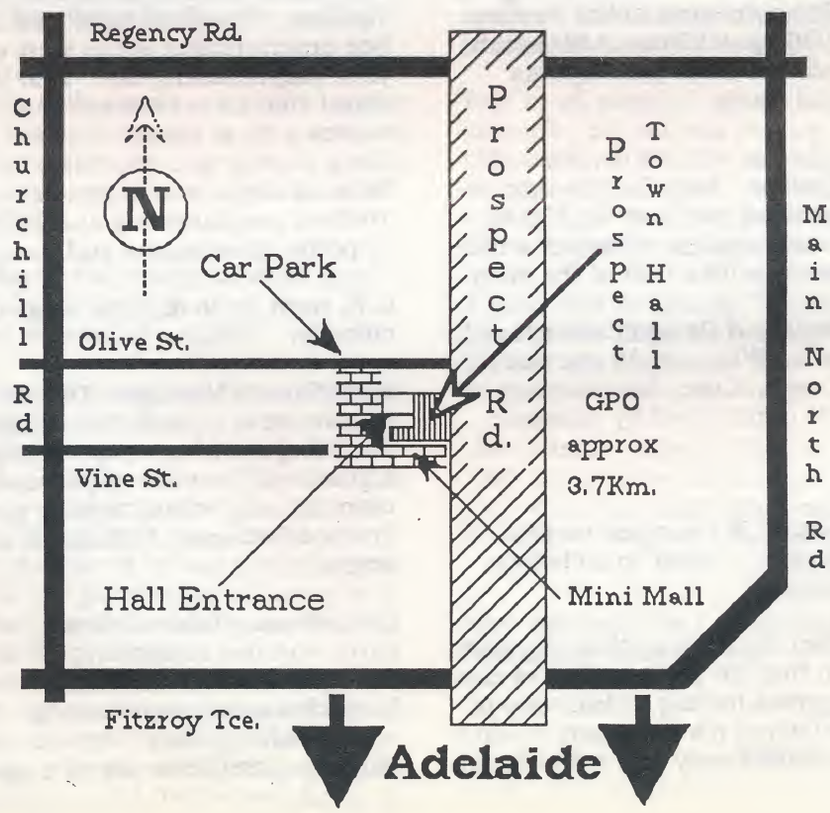
Macintosh Meeting
Friday 16th November 1990 at 7:30 p.m.

- Poor man's introduction to Animation using Pascal, SuperCard and Super-Paint - by Michael Coleman.
- Question and Answer section
- Boxes of 3.5" disks for sale
- Macintosh public domain disks for sale
- Free tea, coffee and biscuits
- All are welcome

Newsletter contributions may be submitted to any club committee member on paper or disk. Articles on disk can be copied onto club disks at meetings. All disks will be returned to you as soon as possible.

Deadline for next Apple-sauce

9th November 1990



Kansas Fest Report #1

Notes from Kansas: Views on the A2-Central Developers Conference by Marc Farnum Rendino

From the Internet

Day 0: The second less-than-semi-annual GS College

The first GS College was in 1987. To give you an idea how long ago that was, Matt Deatherage, Keeper of the General Truths was still in Minnesota then!

This version of the GS College was run by Apple II Developer Tech Support -- a heck of a great bunch of folks for whom the slogan "Apple II Forever!" is as meaningful as ever! They came in force (about 30 folks from Apple all together!) and did a great job.

Only some KansasFest attendees opted to participate, so the atmosphere was generally less like a lecture, and more like a two-way interchange. Levels of experience ranged from beginner to expert and it seemed to me that all were well served.

There were advanced discussion sessions where one could pick the brains of the engineers who actually wrote parts of the System Software (ex: GS/OS, QuickDraw, MIDIsynth, etc.), but it seemed that most participants stuck to the general track.

The general track began with an overview of GS/OS and the Toolbox. Matt Deatherage described the boot process and how GS/OS is able to boot from anything (ex: a server) unlike other operating systems (like that of the Mac).

C.K. Haun (of WriteIt and DesignMaster fame) then discussed tools. Why should one use tools? Reliability, ease of use, less memory wasted, and they're maintained by someone else!

Next Eric Soldan and C.K Haun led us into programming strategies -- what to do before you get to the keyboard.

Eric's prefers to plan the code so that bugs are avoided, and when they do pop up, they're easier to find. He suggests having a clear view of the end result, and when it's necessary to vary from that, to do it consciously and with consideration.

Another good suggestion offered by Eric is to plan for when to freeze the system and stick to it! For example, when a new System Disk comes, it's very easy to fall into the trap of integrating some of the new features. Don't! Ship it, and then work new features into a new release. A spec that's constantly changing is one that doesn't ever ship, and if it does, it's not too solid.

Hints for planning the job:
- o write the way you think
- o design it to debug it
- o maintainability and expansion -- the world changes!
- o segmentation
- o use resources!
- o keep "time to market" in mind:
  think of it as a race; there's only one winner!
  choose a path and stick to it (not backtracking helps!)

C.K. Haun then expanded on the use of the Toolbox. "Tools are your best buddy." "Toolbox programming is the start of the rest of your programming life." And if you think about that for a brief moment, you'll realize it makes a lot of sense! Toolbox programming is like a lead-in to object oriented programming. In some ways, it is object-oriented. Surprise! Toolbox programming also helps tremendously in ports, conversions and parallel development.

C.K. went on to discuss some of the tools specifically:

The Memory Manager: Use it! It works well! It takes care of a lot of the drudgery for you. And don't forget the new Out Of Memory Queue; it's a great way for your application to be much more friendly when memory gets tight! (See Toolbox Reference Volume III and the Technotes.)

QuickDraw: Take advantage of "fast port" features and use shadowing. Follow the rules! Doing so allows you to take advantage of Jim Mensch's constant tweaking. He sped things up something like 4 times for System 5.X; he says it's possible to speed it up even more!

Continued next page

The Window Manager: TaskMaster! And now TaskMasterDA! Make your content draw routines smarter. Check the port. (General clue: If you're not passing the port to a call, do a GetPort and SetPort before it -- it'll save some headaches.) RefCons: Are you wasting them? They're there just for you!

The Control Manager: Use extended controls! (Otherwise TaskMaster can't handle it for you.) Use custom controls.

The Resource Manager: It's not a replacement for a data file! It does take a lot of the management burden off you, by taking care of handles, memory, disks, etc. It is possible to use more than one resource fork...

Eric expounded further on resources. "Data is easier to debug than code." Think about it!

More on design ("Starting not from scratch"): Design the first application to be recyclable. ("It's good for the programming environment." <grin>) This will probably save time for the first app, and definitely save time for subsequent ones. It generally produces better code, since it's already been tested!

Your app is similar to something -- start there! If you don't have one that's similar, someone does. Like DTS -- they have several megabytes of sample code!

Eric then demonstrated the idea of using one app as a base for a new one. One reason he chose the one he did was that it was fairly recent, so he was familiar with it.

A few last tips from Eric: Use a debugger to trace through the code once -- you may be surprised what you find! Also, get someone else to look at your code, since it's very easy for you to read what you think, rather than what's actually on the screen!

After lunch we faced the reality of what happens when you do sit down to write something: it crashes. <grin>

Eric Soldan: Worry about speed later. Besides, speed improvements are often necessary in exactly the opposite places you thought. Use debugging code like "PtrCheck", a utility that Eric designed to make sure you're not using uninitialized pointers. (PtrCheck is available from DTS.) Code so it's difficult to use something incorrectly. Check the parameters on entry to a routine; don't assume that since you wrote it you'll always use it correctly.

Then Dave "Mr. Tangent" Lyons stepped in. He works a little differently from Eric. He like to debug! He dumps into the monitor on purpose to see what's going on and, before you know it, he's tracing some system software totally unrelated to what got him into the monitor in the first place. (Hence the nickname.) His idea is that he'll probably end up needing to know about it sooner or later, so why not trace it as it comes up!

Dave is the author of the famous Nifty List CDA which is the first thing any programmer should get. Or anyone who's even remotely curious about the innards of the system. (DAL Systems, PO Box 875, Cupertino, CA 95015-0875) Dave gave attendees version 3.0 which, in addition to all the things Nifty List is famous for, now adds things like "Big Brother" which monitors your memory use. ("Hey! That's not your memory!") If you don't have Nifty List, get it. Get it now.

At 3 PM Llew Roberts described his experiences in Testing. Know what you are testing; lay out the requirements that the application must meet and design how to comprehensively test them. Design it so that a naive user could follow it -- as a matter of fact, it's better if a naive user actually performs the test. You don't want bug reports, you want the tester(s) to find them before you ship.

Design the test plan from the specs, not from the application itself. Test the limits of the program. Find places where there are constraints, such as a routine where a date is read in. Test for values that are outside the range of valid data. Test to make sure that defined functions are indeed performed properly. (Does is print?) Then the fun part: destructive testing. Give it to your mother and watch what breaks when a real user get ahold of it!

Test in all possible system configurations. For example, unless you require a color monitor, test it with a B&W -- you might be surprised how much suddenly becomes invisible or indecipherable. Try it with old System Software. Don't forget about floppies!

A good rule of thumb: "A hack today is a crash tomorrow!" Then there's regression testing. Try out old bugs. Are they really fixed? Did the fix introduce any new bugs?

Use outside testers. They're honest! <grin> They have no vested interest and they'll tell you when they don't like something. And they'll hit things you've never thought of.

Next up were Jim Luther and Dan Strnad to talk about connectivity.

Continued next page

Jim reminded us of the 2nd edition of the IIGS Hardware Reference and Technote #30 for corrections to it, the Firmware Reference and Technote #25 for changes to it, and an upcoming Technote on the IIGS memory slot.

Dan launched into one of the main uses for AppleTalk, Apple's File Server software, AppleShare. Some Dos and Don'ts for AppleShare: let the user know of his/her privileges, temp files must have unique names since other users may be storing temporary data to the same place, minimize segmentation to avoid having to go back to the network many times (for speed).

There are also two calls that may come in handy for when you need a bit more time than AppleTalk is willing to give you between checking that you're still there: RAMForbid causes AppleTalk packets to be buffered and RAMPermit turns buffering back off. (See the Technote #3.)

Then Jim got into network printing. There are basically four different ways to address AppleTalk printers! Via the Toolbox Print Manager, the Remote Printer Manager (RPM), the .RPM GS/OS device driver, and the Printer Access Protocol (PAP). Start at the top of the list. If you can't find the service you need, move down the list. For example, it may be necessary to use PAP to download LaserWriter fonts, but printing a document to the LaserWriter can usually be done with the Print Manager.

Day 1!

The day started with the "Apple IIGS System Software Update" which was basically a repeat of the session in May at the Developers Conference. There was also a session on some new happenings in the ProDOS 8 world, but I was busy at the IIGS session! I hear that there will finally be a selector built into ProDOS 8.

The next session was with Eric Soldan describing Dynamo, a system that adds a great deal of power and flexibility for the ProDOS 8 developer. The code generated with Dynamo is often smaller and faster than assembly! Which makes a bit of sense because it actually is assembly that's been tweaked.

Recent additions to the Dynamo package include integer and floating point math, string handling, and arrays. All this and it uses only two bytes of zero page!

When you assemble the source you get to experience the other side of the Dynamo package: AppBuilder. A powerful linking/loading system that can load the code anywhere you want: any bank, the language card, whatever. And overlays too!

Dynamo is available in an MPW version and an APW version. Due to some problems with APW macros, the macros of Dynamo are not available to APW users. Though a good portion of the system, Dynamo is still very useful to the native developer.

(At the same time Mark Cecys demonstrated MIDIsynth and synthLab to an enthralled audience. It really has to be heard to be believed!)

Jim Luther and Dan Strand discussed AppleTalk further, exposing the ability to do asynchronous calls. In effect, telling AppleTalk to do something, and getting immediate control back; before the task is completed! This could come in handy! Even for such mundane things as giving the user the chance to cancel something.

(At the same time Jim Mensch showed off his Animation Toolkit. This tool is amazing and yields at least 30 frames per second (maybe 60?) of multi-character animation. It can be used just like any normal toolset (I believe it will be distributed as any other tool; with a system disk), OR you can ask it to generate a jump table for more speed, OR it can generate code for you (and then you can toss the tool out of memory), OR it can simply be used to generate characters! And that doesn't even cover it all!)

At lunch we got to meet Jane Lee, in Apple II Product Support. She's very personable, knowledgeable about the Apple II, and won't take no for an answer! It recently came to her attention that there was going to be a major Apple ad in the Wall Street Journal, but only the Mac was in the ad. After she was done with it, it mentioned the Apple II several times and had some Apple II pictures in it. We're lucky to have her on our side!

She told us that most Apple II's are purchased in the school market, but Apple is looking into new markets. There is also work on new advertising and Jane is trying to get some money for TV ads! Also, work will continue to insure that the Apple II is well represented in generic Apple ads. Apple II Evangelism and more R&D are also being worked on!

After lunch we signed a non-disclosure agreement that for System 5.0.3 and HyperCard GS... All I can say about it, is that it looks solid and that this will not knock off HyperStudio. HyperCard is a programmers tool; HyperStudio is a users tool. HyperStudio remains the most accessible way for anyone to use an Apple and all the myriad things that can be hooked up to it.

Continued next page

At dinner we got to meet another of our supporters at Apple; Ralph Russo who heads Apple II Products. He is a dynamic man, and is devoted to fixing the Apple II situation. He's been on the job less than 2 months, but already things are jumping! He hopes to have a strategy laid out before summer's end. As far as general plans go, he agrees that it's necessary to be aggressive; not to sit back on the "continue to service and support" line. Our kind of man!

Day 2:

I began Saturday with Matt Gulick regaling us with tales of SCSI. He told us about what the High-Speed SCSI card and his new drivers can do. Did you know that with these drivers you can hook up virtually any hard disk? And it'll work! This goes for any fixed drive (like a Seagate) _and_ for removable media like SyQuests and optical drives. Just plug and play! The drivers ask each device what it's capable of and react accordingly.

And he showed us some new things like scanner and tape drivers. With an application that a tester whipped up, he showed us some _amazing_ scans. The quality was astounding! And, just because it can be done, he scanned intonormal HiRes! It wasn't half bad! (And the app had been written overnight!)

We also got to see the "Star Wars" demo. (Actually, The Empire Strikes Back.) In this demo, digitized (from a laser disk) frames of the movie are played back from a hard drive to the Super HiRes screen of the IIGS -- faster than real time! And it's done with one SCSI call!

The new SCSI card supports full arbitration. For example, it would be possible to tell two devices to do something to each other, send them on their way, and get control of the CPU back for something else!

At 10 AM I ran a session on Computing for the Handicapped. We didn't fill the room, but we did discuss things from the Adaptive Firmware Card to the Human Interface itself. Don Brady from Apple was nice enough to join us, and I'm sure he's already talking to the right people about some of the extensions we'd like to see!

(At the same time were sessions on CD ROM and the Apple II (Llew Roberts who helped with the GEM CD) and GS Programming in the 1990's (Tim Swihart). I hope others who went to these sessions will jump in to let us know what went on there.)

Next was Media Integration with Dan Hitchins and Jim Merritt. We got to see some amazing things in this session. For example, there is a product from Houghton Mifflin called "EcoVision" which is based on a video disk and uses a sort of a hypermedia interface to get students to think about, research and report on environmental concerns in their area. There are live video clips and even a video "agent" to guide the student. It really is amazing!

Also, we were shown some demos of a 640 X 400 resolution screen! Using technology that's readily available today --
the Video Overlay Card! It's not at all trivial to get the system software (like QuickDraw) to use this, but one wonders if we might see this sort of thing used in future systems...

At lunch we met David Szetela who is head of Apple Developer Support. You remember David from his Nibble days don't you? <grin>

One of the neat things David had for us was the revelation that Apple II developers (not Mac developers) from outside the US will now receive support direct from DTS in Cupertino! There is a new direct DTS fax number as well!

David also discussed the new Partners and Associates programs where an Apple II developer can get into the game for as low as $150! What are you waiting for folks?! Jump in! Believe it or not, this price also gets you hardware discounts and an invitation the yearly Developers Conference! It's finally feasible to show Apple how many of us Apple II developers there really are. Imagine the reaction at Corporate if only half of us joined up?!

Later that afternoon Matt Gulick ran another session that got down and dirty about SCSI. (Sorry!) We learned that it's possible for a driver to be loaded after boot, but not possible to unload it. There is also a relatively new GS/OS call that allows anyone to flush the cache ($2026).

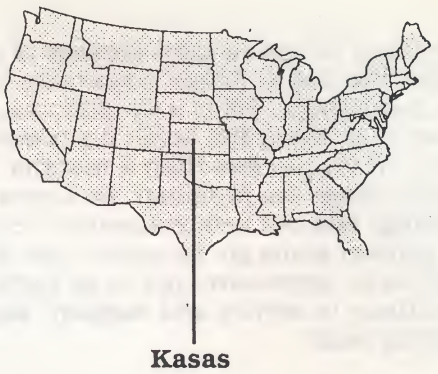
And Eric Soldan went into detail about printing. Apparently printing is the last thing people think of, and it's not all that simple. One bit of sage advice that he had to offer was to keep printing in mind when writing window content draw routines -- it can make printing a lot easier.

Tired, but happy we thought about going to the baseball game, but it was not to be! All hell broke loose and we all got soaked. So we went to "Crown Center", watched a movie (a group of us saw DieHard II), scared the heck out of some poor hostess at a restaurant downstairs, and took over the back room until the bus showed up.

Continued next page

All in all, one heck of an event! Tom Weishaar is to be thanked! As well as all the great Apple folks that came! And all the other folks, from the stars to the unknowns! (I think some of the unknowns can't claim that title anymore!)

I wonder if Tom is taking reservations for next year yet...



Ducks

Bill Currie

Reprinted with permission from AUSOM October 1990

When this program runs you will see two ducks fly from water level to the top of the screen.

As usual, I have provided a one-line version, and a longer listing for those who don't want to try cramming it all into one line.

```
1  FOR S = 0 TO 7: READ A: POKE S,A: POKE 232 + S,0:
   DATA 1,,4,,2,4,5,: NEXT : HGR : HCOLOR= 3:
   FOR X = 0 TO 279:Z = Z +  RND (1) * 3 - 1:
   HPLOT X,0 TO X,190 - Z: NEXT : FOR C = 0 TO 9E9:
   FOR U = 159 TO 9 + C STEP  - 1:F = F = 0: FOR D = 0 TO 1:
   SCALE= U / 9: ROT= 8 + 32 * F: XDRAW 1 AT U,U:
   XDRAW 1 AT U / 2,U - 9: PRINT "~^";: NEXT
D,U,C,K
```

```
1  FOR S = 0 TO 7
 2  READ A
 3  POKE S,A
 4  POKE 232 + S,0
 5  DATA 1,,4,,2,4,5,
 6  NEXT
 7  HGR
 8  HCOLOR= 3
 9  FOR X = 0 TO 279
10  Z = Z +  RND (1) * 3 - 1
11  HPLOT X,0 TO X,190 - Z
12  NEXT
13  FOR C = 0 TO 9E9
14  FOR U = 159 TO 9 + C STEP  - 1
15  F = F = 0
16  FOR D = 0 TO 1
17  SCALE= U / 9
18  ROT= 8 + 32 * F
19  XDRAW 1 AT U,U
20  XDRAW 1 AT U / 2,U - 9
21  PRINT "~^";
22  NEXT D,U,C,K
```

Speech delivered at National Apple User Group Conference, April 21, 1990, Chicago, Illinois, by Rajiv Mehta.

1) Hello, as Mark said, I'm the product manager for the Apple II and I'm here to talk to you about the Apple II as well as Apple's overall strategy for entry-level products. I signed up for this quite some time ago, when I first learned about this conference, because I figured it would be a great opportunity to discuss these issues with some of our most knowledgeable and influential users.

2) Later on, sometime in March, I learned that the other speakers were Bill Gates, chairman of Microsoft, and Mitch Kapor, chairman of On Technology and founder of Lotus, and Guy Kawasaki, of Apple and Acius fame. My first reaction was "What a deal! They said you could rise fast at Apple, but this was unexpected ... sharing billing with some of the computer industry luminaries!" However, this euphoria was soon followed by "Oh no! What have I gotten myself into? It sounds like if these big guys are going to be there, someone with a more impressive title should be here from Apple." So, I frantically called around to see if John Sculley, Mike Spindler, or other appropriate heavy-weights, could do this instead. All the replies I got were the same: "sounds like a great event ... can't do it ... sure it's in good hands!"

3) I tell you this because it's a clear example of the type of company we are. Everyone gets involved, everyone has a thing to say, and everyone deals with our customers. This also means, as most of you are well aware, that we often operate in a state of marginally-controlled chaos, a creative anarchy. This environment is what attracts so many of us to Apple in the first place. It is very much a part of the enthusiasm and energy that we have for what we do — for our products, and you, our customers.

4) And from day one, this same energy has been apparent in you as well. The level of loyalty and enthusiasm that both Apple II and Macintosh users have for their products is completely different from the emotions generated by anything else we use. There are times which make me wonder whether we want such enthusiasm: after I've dealt with a particularly irate customer; or listened to people who have very strong opinions of what we're doing completely wrong (and who have no hesitation in explaining this to me in complete detail); or whenever I first hear of the latest rumor. Most of the time however, particularly when I get to visit our customers at schools and user group meetings, I realize that your enthusiasm is why I work at Apple. And that's why we're here, to address some of the issues that you tell us are of greatest concern to you.

5) Identify confusion Let's start off by identifying these issues. Lately the level of rumors has risen to a feverish pitch, from those that say that the Apple II is dead and that Apple is completely abandoning the Apple II community, to those that speak of a glorious revival of the Apple II and claim to have complete details about all these products that we're supposedly working on. There are also people who claim that Apple doesn't care any more about the individual user or the education marketplace — basically that our goals and ideals aren't what they used to be. Well, what I'm going to try to do today is set the record straight, to clear all this confusion. I decided to do without any splashy demos, Jean-Luc will have plenty of these tomorrow, what I have to offer instead is some straight talk.

6) Extract non-issues out of confusion Let's first deal with this notion that the Apple II is dead. There are over 5 million Apple II computers out there. There are hundreds of thousands of schools, small businesses and individuals who use them every day. The Apple II continues to be the foundation of K-12 instructional technology in schools throughout the nation, and we expect them to keep using these machines well into the 1990s. In the past two years alone, there have been about a thousand new educational software packages for the Apple II. There have also been wonderful products such as HyperStudio from Roger Wagner, GTV from National Geographic, and a whole assortment of hardware from Applied Engineering, just to name a few. And in the past year, we've introduced the video overlay card, system software 5.0, a new IIGS with 1 megabyte of memory standard, and most recently the high-speed SCSI card.

7) We have also said many times that we expect to be making the Apple II well into the 1990s. I think it is fair to say that the Apple II is not dead; rather it is alive and well.

Continued next page

Speech delivered at National Apple User Group Conference, April 21, 1990, Chicago, Illinois, by Rajiv Mehta.

1) Hello, as Mark said, I'm the product manager for the Apple II and I'm here to talk to you about the Apple II as well as Apple's overall strategy for entry-level products. I signed up for this quite some time ago, when I first learned about this conference, because I figured it would be a great opportunity to discuss these issues with some of our most knowledgeable and influential users.

2) Later on, sometime in March, I learned that the other speakers were Bill Gates, chairman of Microsoft, and Mitch Kapor, chairman of On Technology and founder of Lotus, and Guy Kawasaki, of Apple and Acius fame. My first reaction was "What a deal! They said you could rise fast at Apple, but this was unexpected ... sharing billing with some of the computer industry luminaries!" However, this euphoria was soon followed by "Oh no! What have I gotten myself into? It sounds like if these big guys are going to be there, someone with a more impressive title should be here from Apple." So, I frantically called around to see if John Sculley, Mike Spindler, or other appropriate heavy-weights, could do this instead. All the replies I got were the same: "sounds like a great event ... can't do it ... sure it's in good hands!"

3) I tell you this because it's a clear example of the type of company we are. Everyone gets involved, everyone has a thing to say, and everyone deals with our customers. This also means, as most of you are well aware, that we often operate in a state of marginally-controlled chaos, a creative anarchy. This environment is what attracts so many of us to Apple in the first place. It is very much a part of the enthusiasm and energy that we have for what we do — for our products, and you, our customers.

4) And from day one, this same energy has been apparent in you as well. The level of loyalty and enthusiasm that both Apple II and Macintosh users have for their products is completely different from the emotions generated by anything else we use. There are times which make me wonder whether we want such enthusiasm: after I've dealt with a particularly irate customer; or listened to people who have very strong opinions of what we're doing completely wrong (and who have no hesitation in explaining this to me in complete detail); or whenever I first hear of the latest rumor. Most of the time however, particularly when I get to visit our customers at schools and user group meetings, I realize that your enthusiasm is why I work at Apple. And that's why we're here, to address some of the issues that you tell us are of greatest concern to you.

5) Identify confusion Let's start off by identifying these issues. Lately the level of rumors has risen to a feverish pitch, from those that say that the Apple II is dead and that Apple is completely abandoning the Apple II community, to those that speak of a glorious revival of the Apple II and claim to have complete details about all these products that we're supposedly working on. There are also people who claim that Apple doesn't care any more about the individual user or the education marketplace — basically that our goals and ideals aren't what they used to be. Well, what I'm going to try to do today is set the record straight, to clear all this confusion. I decided to do without any splashy demos, Jean-Luc will have plenty of these tomorrow, what I have to offer instead is some straight talk.

6) Extract non-issues out of confusion Let's first deal with this notion that the Apple II is dead. There are over 5 million Apple II computers out there. There are hundreds of thousands of schools, small businesses and individuals who use them every day. The Apple II continues to be the foundation of K-12 instructional technology in schools throughout the nation, and we expect them to keep using these machines well into the 1990s. In the past two years alone, there have been about a thousand new educational software packages for the Apple II. There have also been wonderful products such as HyperStudio from Roger Wagner, GTV from National Geographic, and a whole assortment of hardware from Applied Engineering, just to name a few. And in the past year, we've introduced the video overlay card, system software 5.0, a new IIGS with 1 megabyte of memory standard, and most recently the high-speed SCSI card.

7) We have also said many times that we expect to be making the Apple II well into the 1990s. I think it is fair to say that the Apple II is not dead; rather it is alive and well.

Continued next page

9) Our primary goal for Apple II product development is to enhance the performance and functionality of the huge installed base of Apple II computers. We strive to make our products compatible with the machines that exist today; to ensure that as many people as possible benefit from the advances in technology that are possible. Every user of a IIe or IIGS can make use of the video overlay card, and the new SCSI card. The software that accompanies the new SCSI card, just to give you an example of the detail we go to, is backwards compatible, so that people using both new and old SCSI cards don't have to worry about two sets of software. System software 5.0 brought a tremendous increase in performance, and provided new networking capabilities, to everyone with a IIGS; not just to those that bought the new one.

9a) For the future, you can expect that our goals, our product development philosophy, will remain the same: building products that enhance the installed base. With such a huge installed base, we believe that this is the best and most responsible way to serve our customers.

10) Keep in mind that the Apple IIe and IIGS are de facto standards in homes and in schools. The value of this standard is that developers know that they can expect everyone to have the same capabilities and know how large the audience is for their products. Those of you who have been using an Apple IIe for a long time will remember when we came out with the 128K version. It was received very warmly; but it was a long time before people started taking advantage of the extra memory. Developers developed programs for the 64K version because that's what they could expect. Similarly, one reason for the slow development of IIGS-specific software is its IIe compatibility — developers can reach more people if they write software for the IIe. Given this situation, it would be a disservice to both our users and developers to make some disruptive change to the Apple II line — something which would be incompatible or not accessible to the vast majority of people who already have Apple II computers.

11) Talk about our product direction This is not to say that we're going to necessarily just maintain status quo with Apple II technology. This is powerful technology and we are well aware of the fact that we're pretty good at it. When and if we find new and innovative things to do, that provide dramatic new capabilities for our customers in a cost effective manner, we will do it. Today, we are actively investigating several projects.

12) At the same time, it should be clear to everyone that we're aggressively working to bring Macintosh technology to many more people. We are committed to bringing the power that's in today's high-end Macintoshes down to lower price points. This is consistent with the way things usually happen — new technologies first appear on high-end machines, and then come down to entry-level systems. I should point out however, that even though a technology often appears on an Apple II or a low-end Mac a little while after it appears on a high-end Mac, it still usually gets there before the rest of the industry! For example, the Apple II has had a SCSI card since early 1987, while IBM has only discovered SCSI today, and only for their high-end machines. We also do introduce new technologies anywhere in the product line when it is appropriate — the Apple II video overlay card is a great example of that.

13) However, there does seem to be a lot of misunderstanding of what our intentions are. Back in January, I met with Jerry Cline of the Arizona Apple user group, to ask his ideas on the Apple II situation. As a follow-up to our meeting, Jerry asked Apple II users to provide me with their "wish list". The response to that was a little overwhelming, but after sorting through the list, one of the things that came through loud-and-clear was an insistence on this emotionally charged separation between the Apple II and the Macintosh. There were a lot of people asking for a more powerful Apple II. Some of these people even told me to remove the IIe capability from the IIGS! And they listed all sorts of features that should be standard. The ironic thing is that almost all the features they were looking for already exist on various Macintoshes. However, they apparently won't consider these products because they have the wrong name! This is really silly! Apple II people give grief, justifiably, to Apple when we make the mistake of pushing Macintoshes to people that really would be better off with Apple IIs. At the same time, people that use the Apple II today should be open when they're suggesting to other people what product is best for them — take a look at our whole range of products. We really want people to think of our whole product line, the Apple II and the Macintosh, as a wide range of choices, all of which may be suitable for their needs. We don't want our customers to be pigeonholed into separate Apple II and Macintosh camps.

14) We also understand that many of our customers use both Apple IIs and Macintosh in the same environment, and we have been helping them do this for a vey long time. I'd like to point out that the Macintosh is the most "Apple II compatible" computer that exists today, and vice-versa. We have built many "bridges" between the two products. Just consider the following:

Continued next page

22) Ask for help in fixing these We solicit your suggestions in developing ways to improve these situations. I, along with the many other people from Apple that are here at this conference, have been talking with many of you to discuss possible programs and solutions. The discussions yesterday, in the late-night rap session, and earlier this week at the User Group Advisory Council meeting were very valuable. These are difficult issues, and we very much appreciate the help that you've provided us, and we will continue to count on your help.

23) Ideological debates So far, I've dealt with the state-of-the-Apple II, about what we're doing with Apple II product development, about "bridges" between our product lines, and what some of the issues are in the service and support arena.

24) To close, let me deal with this notion that we're not the same Apple, that our goal of producing computers for everyone, for the common man, have changed. There are generally two ways this comes up in the Apple II community, and they're actually very related. They are that the Macintosh is a machine that is not conducive to tinkering, and that we impose too many strict regulations about how software should work.

25) What got Apple started in the first place was the idea that Steve-and-Steve had to bring what they loved to more people. While the members of the Homebrew Computer Club were all perfectly capable of putting together a bunch of assorted parts and assembling a computer, most of the world wasn't. Apple started because we wanted to bring this wonderful new tool to everyone, to take this technology which is our passion and to turn it into something that helps each and every one of us change the world. Everything we do is aimed at making this technology accessible to more and more people.

26) In my student days and in previous jobs, I took pride in writing a simple plotting program on my Apple II back in 1980; in building a little computer, power supply and all, to run my model train set; and in creating a log-on file in my VAX account so that no one else could possibly use my computer. However, people like me, and the many Apple II hackers I've met in user groups, who can get excited about technological details and find it challenging and fun to make the computer do stuff it wasn't really meant to, aren't the norm — most people aren't like this — they don't want to have to know this stuff, they just want to get their work done. What we at Apple want to do is to make it possible for both groups of people to have a choice, to do more, and to have a lot of fun in the process.

27) Similarly, the human interface guidelines we have for Macintosh and IIGS programs are well suited to the vast majority of users. Some Apple II developers complain that these guidelines cramp their style, that they don't allow them to be creative, that they know of better ways to do things. All I can say is: the success of our interface, the fact that the entire industry is trying to be "Mac-like", proves that this is a good idea, that this is what people want. There is of course lots of room for improvement, and there are lots of developers and users who provide us with invaluable feedback and ideas to make things even better. Their creativity is definitely valued. But the idea is to build upon what we already have, to make things better, not to start from scratch every time.

28) Some people say that the days when a lone individual with a great idea could create a program that would make a big impact are long gone. They say that people like Bill Gates and Mitch Kapor would just not make it today. They say that because programs have gotten so big and complex, an individual can no longer write something ground breaking. We think this is nonsense, and we are working hard to ensure that anyone anywhere with a crazy idea and a lot of passion can still "change the world". We do this by giving them more and more powerful tools. The system software suites of the IIGS and the Macintosh are just that — powerful tools on powerful platforms to build upon. Just ask Roger Wagner how much harder it would have been to do HyperStudio if he didn't have our tools to start with.

29) I'm an amateur photographer and I can tell you about a similar debate in photography that happened about a century ago. At that time, photographers used a wet-plate negative — basically you mixed up a batch of chemicals, coated a plate of glass with this goop, and quickly put this into your camera to take the picture. A basic requirement for being a good photographer was being a good chemist. Then along came the dry process — the kind film we use today — and George Eastman got into the business, selling film. A lot of traditional photographers looked down on this whole thing; they felt that they were giving up control to a company, Kodak. While true to some extent, they completely overlooked three very important facts: one, that they could now concentrate on the art of photography and not the craft of mixing chemicals; two, that photography was made accessible to far more people; and three, that it became possible to photograph things you could never do before (imagine trying to take candid photos when it took you twenty minutes to get the camera ready).

Continued next page

30) In the same way, what we're doing is making computing accessible to far more people, and making it possible to do much more with computers than was imagined just a short while ago. We do appreciate the skills and talents of those who know by heart the memory map of a IIGS or who have mastered the use of Peek and Poke in Applesoft BASIC — we just want you to do even more with these skills.

So, I hope that you will go away reassured that Apple has not lost its way; we're remaining true to our roots, and I hope that you will continue to find the journey as rewarding as we do.

31) Thank you.

Downloaded from America Online
MUG NEWS SERVICE, 1990

Reprinted from Macinations, September 1990

Windows 3 – No Threat To Mac

by Grant Gillespie
Editor of MAQuill, The Magazine of the Macintosh Association of Queensland

YOU may have seen or heard a lot in the press and on the grapevine about the new IBM environment called Windows 3. Essentially, it puts a glossy "windows" veneer on good old-fashioned DOS. The difference is that with version 3 it actually **looks** good – believe me, there are windows and icons for everything! And this is not only it's strong point, but also it's downfall. "You want windows? We got windows galore."

Apparently someone told Microsoft (you know – the people who wrote that Mac masterpiece MS Word) that people like to use windows and icons. What got lost in the translation is that "people like to use windows and icons **if and only if** they are easy to use!" And anyone who has grappled with Word on the Mac can tell you that the intellect behind it could not create anything intuitive or easy to use... if its very life depended on it. Sadly, with Windows 3, this has happened again.

Having watched people struggle to find their way around Windows 3, I can say that the old ethic of "the harder it is to do something the better it must be" is alive and well.

This is no 'Mac on an IBM'. It is as cumbersome and difficult to work with as DOS, with the added complication of masses of icons and windows completely disconnected from the file system. Microsoft has somehow managed to give the graphical user interface (GUI) a bad name. No wonder Apple held them up in the courts all this time. It seems that mongrel Apple was simply trying to protect the GUI from people who don't even think its a good idea and are only making a GUI now as a marketing exercise.

The reason the Mac is so good, is not merely because it has windows and menus but because a lot of research goes into finding out "how people would like to use the machine" instead of "making the people fit the machine". The Mac was designed with its GUI actually in the machine (programmers call this the ToolBox) and the result of this is that the machine matches the person. Windows is an add-on, an after-thought, and this shows when you come to use it (as opposed to watching the impressive demos).

And impressive they are. But the reality is that Windows needs 5 Megs of disk space (this is just the environment which sits on top of the operating system, remember) and requires a minimum 386 with 4 Megs of RAM to do what the Mac can do in 1 Meg of disk space and 2 Megs of RAM. And this is only the beginning.

Here are some further examples:

- The "work group" idea is clever – grouping applications and files together in your own "project" window – but if you run one of these programmes and save a file, it doesn't appear back in your "work group", you have to go chasing through the filing system and put it there!
- And you don't realise how elegant the Mac is until you have to run two windows sessions of the same programme, just to get two documents open so that you can cut and paste between them. And this you can do in Finder! To even attempt to compare this clumsy set-up to the intuitive, simple MultiFinder is a joke.
- ToolBook looks very much like a glossy, full colour HyperCard. But the reality is that it runs too slowly to use, even on a 386. (HyperCard is quite acceptable on an 030, albeit too slow on a Plus).

There will always be people who value *money* over *time*, and don't mind taking forever to learn how to get the computer to run the new programme. But for those people and businesses who value *time* over *money* and just want to get on with work, there will always be Macintosh.

TypeStyler

Reprinted from Macinations, September 1990

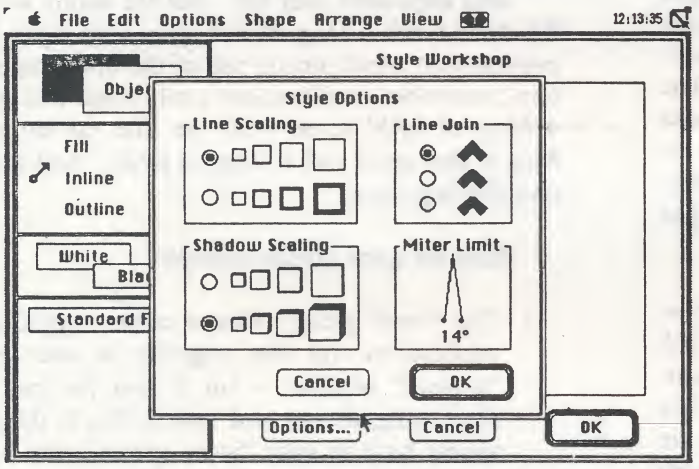
by Daniel J. Hanna

OUTSIDE of entertainment software, Broderbund is relatively unknown (even though it's been around since the Apple II). TypeStyler is an enjoyable program and a useful one from Broderbund, aimed at providing varied text effects to the home or amateur desktop publisher.

Early attempts at making headings and poster text look exciting on the Mac were flawed in several ways. LaserFX by Postcraft International created some incredible Postscript effects, many of which approached professional quality. The downside was that the program crashed (a lot); it didn't even run properly with 1987 system software, let alone System 6. Being Postscript-based, the results were useful only to LaserWriter owners.

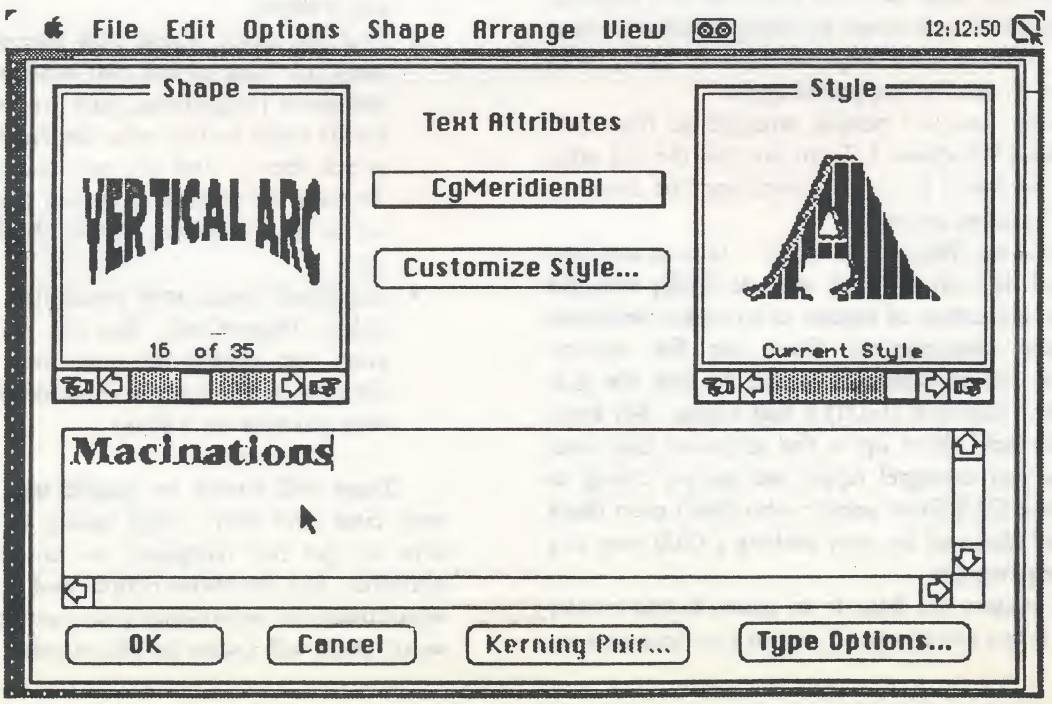
TypeStyler takes a broader view. The application is like ATM and LaserFX rolled into one, meaning that you get good-quality output on anything from an Imagewriter up. It comes with several proprietary "SmoothFonts" (licensed from Agfa Compugraphic), although you can use any Postscript font with its built-in Font Converter utility. TypeStyler uses these fonts to bend, twist, shape and fill text in all manner of ways.

And it really works! I found it stable even with my worst-behaved INITs. On a Mac Plus with 1MB of RAM it was pretty slow, but the SE/30 handled the maths much more quickly. In either case the results were worth the wait, and were surprisingly good on the Imagewriter.



There are enough options in the program to keep even a power user happy. Letter-spacing and several types of fill and shadow are available, all of which are viewed on-screen in WYSIWYG. Files can be exported to and imported from Postscript, EPSF, Illustrator, PICT and Paint formats.

The impressive flexibility of options has a price, however. The final output is not always as nice as the older, more specialised programs like LaserFX. But with a snazzy user interface, good programming and (above all) an opening for my creativity, I could be prepared to compromise. MAC



Inside the MacIntosh.

By Ian Bagust.

In an article on the basics of a computer it is not that easy to know where to start and how deep to go. Some will think its too basic and others not enough details, but here goes anyway, the primary aim is to improve the understanding of the MAC, for beginners, without too much detail.

A computer consists of a number of parts.

1. The CPU.
2. The Memory (ROM and RAM)
3. The input devices (keyboard etc.),the output devices (display and ports).
4. The Power Supply.

The CPU.

1. The MacIntosh uses the Motorola 68000 series CPUs i.e. the 68000 for the original to the Mac Plus and SE. The MAC-II series started out using the 68020 but has changed over in later models such as MAC-IIcx etc.to the 68030 which is also used in the SE/30.

The CPU is where the actual computing is usually done, in that data is taken from the input device or memory and manipulated in some way and the result transferred out to another memory location where it may be sent out to an output device. All this takes place at a very high speed; in the case of the 68000 it is being clocked at 8 Mhz. (Later machines with the 68030 at 16Mhz up to 40Mhz for the MAC-IIfx) A CPU has address and data lines plus control lines for controlling its operation. The 68000 is a 32 bit microprocessor internally, but uses 16 bits externally so has 16 data and 24 address lines.

The actual CPU is controlled by what are called machine code instructions and these several hundred instructions are used to perform various functions, e.g. One instruction may tell it to take a number from a certain memory location and put it in the internal register, the next instruction may say take another number from a different memory location and add it to the number in the register, the next instruction would be to put it in another memory location or send it to an output device. All this would take place in the space of several microseconds (millionths of a second). So what each instruction does may be relatively simple but because of the speed, a lot of repetitious work can be done in a very short time, which is really all a computer does.

Memory (ROM and RAM)

2. Ram and Rom memory in the 68000 series is 16 bits wide each bit is capable of holding only two states (on or off) i.e. they are binary bits.

There are two types of memory used, ROM (Read only memory) and RAM (Ready access memory). These two perform quite different functions in the computer.

(a) ROM as it says, can only be read and cannot be changed by any action of the operator. In the Mac the original ROM held 64K (64000 bytes of memory) but then went to 128K for the Plus and to 256K for the SE onwards. The ROM is the source of all the instructions for the computer to start working. It has a large number of procedures which are accessible to programmers writing an Application.

The instant a machine is switched on the CPU address register will be reset and it will go to a specific memory location for its first machine code instruction. After this instruction is carried out the register will be incremented and the next instruction will be read and so on. The first instructions will be initialisation and a memory check. Then it will look at the input device such as a floppy disk for the presence of a System file. If the first device does not have a system it will Ian Bagust's article, p2
eject any disk and look at any other device for a system. When one is found it will load it and open up with the desktop.

The ROM has many other utilities available e.g an Events Manager which is constantly looking at (polling) input devices such as the mouse and the keyboard, and if it detects any change, the appropriate action is taken, also if two actions take place nearly simultaneously it will store the second action in a buffer until the first action is finished.

In addition to these features the ROM also holds what are called the QuickDraw routines e.g. if in MACPaint you select the Circle Tool and commence drawing a circle you are using one of the QuickDraw routines. (Of course if a programmer wishes to write a different routine to draw a circle he is able to do so.)

(b) RAM is ready access or read/write memory, where the actual work is done. Firstly the System program is loaded into RAM on start up, then the action starts with first the Desktop being displayed. As we are aware the Desk Accessories and fonts etc. are held in RAM, then any application when it is used is loaded into RAM. The RAM is the working area used e.g. keyboard input of text would first go into RAM, and text in a word processing document would normally be held there until loaded to disk and then lost when the program is closed or the computer turned off.

When the MAC was first introduced, it only had 128K of RAM available but these days 1Megabyte seems to be the minimum requirement and more if multifinder is to be used. The RAM used in the MAC is what is known as Dynamic RAM which means that it has to be refreshed at specified intervals (this is distinct from Static Ram which does not require refreshing) Dynamic RAM generally can be packed into smaller packages and is cheaper. We do not have to worry about the refreshing as it is looked after by the MAC. The important part about RAM is that once the power is removed the contents are lost, whereas ROM data is permanent. RAM in Macs, after the arrival of the MacPlus is held in what are called SIMMS (Single Inline Memory Modules). These plug in devices come in various sizes or configurations and are currently available as 256K, 1 Meg and 4 Meg modules. Although these sizes will no doubt increase as newer, denser chips become available.

Input/Output Devices

The main devices used are;

The keyboard.
The mouse.
Serial Ports.
Disk drives.
SCSI port.
Headset/Audio output.

1. The keyboard is of course our main input to the computer, the output of the keyboard is scanned by the computer at regular intervals, that is, it is an event for the Event Manager. This input is looked at by the ROM or the Application being used and the appropriate action takes place.

2. The mouse input is also very important in the Mac, the position of the cursor or the change in position, and whether the button is up or down are the important events which passed on to the Event Manager.

3. The Mac has two serial ports.One port is used for the connection of other devices such as a Modem where data is received and transmitted in a certain format as determined by the computer, the other port is for use by the Printer or a Network connector. The function of these ports is controlled by the Chooser desk accessory.

4. Disk drives are the source of programs and a means of storing the data which is held in RAM until the computer is to be turned off. The Macintosh started out with drives holding 400K, then 800K and now HD (high Density) drives are used in new machines holding 1.4Megabytes.

5. The SCSI (acronym for Small Computer Systems Interface) port is a high speed port for the connection of devices such as Hard Disks, CDROMS, etc. Each device will have an ID number your internal hard disc will be device 0 and subsequent device numbers 1 to 6. The ID number is set using the Utility program called Apple HD SC Setup for example.

6. Headset or audio output connector is self explanatory. The internal speaker can be replaced by an external speaker to improve the audio quality. Later models have a stereo output plug for stereo audio when available.

The Power Supply.

The power supply is self explanatory and is the means for converting, filtering and reducing the AC supply to the various voltages required by the computer.

Well that's it, I hope that I have not confused anybody with this explanation. If anybody should have any queries please contact me and I will endeavor to help.

Games #21

Space Artillery - Set the angle and velocity to shoot the other spaceship. Beware the stars and a black hole will deflect the missile.
Swamplord - Stop the Real Estate agents from taking over your swamp.
Astro - A computerised astrologer. Very basic.
King Albert - A card solitaire game.
pNukl - Card game similar to Bridge.
Googly Scroogie - A drawing of Scrooge McDuck with eyes moving to and fro.
Jump-It - Jump the pegs to remove them. You win when there is only one left.
Uncle Fred's Nightmare - Boxes bouncing around the screen. Why they thought it was a game when it should be on a graphics disk.
Mr. Lotto - Draws 6 numbers out of 49. Keep drawing until all numbers fall within our 1 to 45 range.
Star Fleet - I not sure the aim of this game, but you try hit the enemy.
Outflank - Another Go type game.
Mr. Cycloid - Spirograph simulator.
Will of Fortune - Loosely based on the Wheel of Fortune TV game.
Time Bomb - A graphic World Builder adventure game. Find the time bomb before it goes off.
Rock Paper Scissors - The old hand game computerised.
Office Attack - Try to mouse click on the moving icons of office equipment to score the most points.
Personal GS! demo - A golf handicapping statistics demo. Enter your golf scores to work out your handicap. Read the demos documentation.
Jan Ken Pon - Crashes on my SE. Could be due to ROMs, or conflict with any of my INITs or CDevs, it may work on yours.

The Font Story
As Easy As A-B-C
All About Fonts - Part 1

by C. V. Rao

For most people, having a choice a fonts is a major reason for using a Macintosh. Afterall, the Macintosh was the first personal computer to have this capability. It sure caught my attention. I was attending night school at The University of Houston when the Mac was introduced and I could just picture how great my schoolwork could look. Word processing with a daisywheel printer was no longer good enough — I soon had my own Mac.

In this series I wish to share my experience and insights with fonts on the Macintosh. We will first look at the evolution of the font system and then get into solutions. My career has taken me all the way from having a Mac as a hobby to setting up Mac equipment for Linotronic Service Bureaus. There were many challenges and I sure saved some tricks up my sleeve along the way.

Continued next page

The History of Font

Today we can choose from thousands of fonts made specifically for the Mac. Eight years ago, when the Macintosh development team was assembled, having a choice of fonts on a personal computer was a significant idea in itself. One did not envision Desktop Publishing, PostScript, Downloadable Fonts, Outline Fonts, NFNT Fonts and on and on. The goal was to keep things very user-friendly, easy to use — keep it simple. Having a menu full of font choices was a dream come true.

So, the first Mac (the 128K) came with less than a dozen fonts — including the seldom used 'San Francisco'. Having only 128K of memory and 400K of disk storage meant having to choose your favorites from those few. To make that possible, 'Font Mover' was included. It allowed the installation and removal of fonts and specific sizes of fonts to and from the System file. Since there were no hard disks, it was simply a matter of configuring several floppies differently to have different font choices. For example, your writing disk may have MacWrite, System Folder and your favorite word processing fonts. Your drawing disk may have MacDraw, System Folder and the neat looking fonts you loved to show off with — keep it simple.

Another reason for this simplicity was the only choice in printers — the ImageWriter. It was not called ImageWriter I — that name came later when ImageWriter II was introduced. (For the scope of this article, they are identical.)

The Imagewriter operates in a very simple manner. For every black dot in your document (text or graphics), it prints a corresponding dot on paper. The resolution of the Mac screen and the Imagewriter are identical - 72 dots per inch (dpi). Of course, the ImageWriter could do better than that — it could do 144 dpi. Things are no longer simple.

Complexity Begins

144 dpi is twice the 72 dpi standard resolution. The Mac screen is, however, fixed at 72 dpi. We are now dealing with multiple resolutions. We need two sets of fonts — one for the Mac screen and one for the "best" quality setting (144 dpi) on the ImageWriter. The solution adopted is to use characters twice the required size, shrink them 50 percent and print them at 144 dpi. When "twice the size" doesn't exist, there is a lot of guesswork. The Mac mathematically guesses the proper location and quantity of the printed dots.

Font Size and Resolution

Throughout this article, only one type of font has been discussed — Bitmap. Also known as Screen Fonts, these type of fonts are designed to work for specific sizes at a specific resolution. To give you an example, let us assume that10-point New York is the only font available on our Mac and look at all the options we have with it.

Using a10-point New York screen font, we can accurately display, on screen, 10-point New York text. On most applications, if a chosen font is available in any of the sizes listed in size menu, that particular size will be outlined (see figure, top of next page). The other choices in the size menu will not be outlined. If we change our 10-point New York text to 14-point New York, the Mac will first look for 14-point New York in the current System file. When it does not find that, an approximate screen 'bitmap' is calculated (mathematically guessed) and shown on the screen. This 14-point New York will be plagued with jaggies. To prevent these jaggies, we need to install 14-point New York in the current System file (see figure, bottom of the next page).

When it comes to printing on the ImageWriter, the very same calculations take place. If we print 10-point New York at the "faster" (72 dpi) setting everything works fine. If we print it at the "best" (144 dpi) setting, the Mac will look for a 20-point New York to shrink to 50 percent and print at 144 dpi. Since we do not have a 20-point New York installed, it simply prints that part of the text at 72 dpi. If we change our text to 14 point and print it, we will get the same jaggies we saw on the screen. The same mathematical guessing takes place and the text is printed at the calculated 14 point at 72 dpi.

We can, however, print 5-point New York at 144 dpi. When we change our text to 5-point New York, our screen image is calculated, since we do not have a 5-point New York available. But, when we print it at 144 dpi, the Mac will look for 10-point New York (twice 5 point), which is available, shrink it by 50 percent and print at 144 dpi.

Mathematical Guessing

In the realm of bitmap fonts, this principle of mathematical guessing applies whenever we use a font or style (yes, style) that does not exist in the current system. The Mac simply adds an extra dot adjacent to all the existing dots in the plain version to get the bold version. Similarly, the italic, outline, shadow or any combination of styles are all created as needed when the right font in the required size and style are not available.

When the Mac calculates what the font should look like (rather then use one as is), the text ends up looking pretty bad — on screen and on print.

Continued next page

Eleven 'Misteaks' That New Computer Users Make

by Lois Silverman, UAUG

1. Buying the wrong computer and hardware. Don't buy the most powerful computer if a smaller model will do. As in cars, there are different models with different capabilities. Determine what you want a computer to do for you. If you are a novice and don't want to take the time to learn, a smaller, more "user-friendly" model might be for you. If you wish to do office work and wish to invest time in learning, a more powerful computer may be for you. Remember that most computers can be expanded to include more powerful features.
2. Thinking one computer will last you forever. Be aware that you may eventually have to trade-up to another more powerful computer as you become more skilled. You would not buy a twelve-speed racing bike to a three-year-old nor would you expect the three-year-old to continue using a tricycle as he grows older. As one's needs grow, a computer will be outgrown and need to be traded-in for a new, more powerful model.
3. Buying and attempting to learn too many programs at once. Don't get overly ambitious since you may become frustrated with the overload. Master a couple of simple applications at first. This will teach you a lot about getting around your computer.
4. Buying new programs rather than using the full potential of those you already have. if you simply operate an application according to what you see in the manual, you are probably missing a lot of capability. Almost every program has hidden features and ways to utilize existing features in imaginative ways. Milk that old program dry before you move on to the promises of new programs.
5. Not analyzing what tasks you will do on your computer before buying software. If you don't, you may buy an overpriced program loaded with features you'll never use in order to do some simple task.
6. Not reading the manual. Many problems may be solved, and time and energy saved, if you read the manuals.
7. Not practicing. Most frustrated computer owners don't practice and play enough. Don't only operate your computer when you have an important task at hand. Practice play a few minutes every day.
8. Not backing up your programs and files. If you copy your applications so you have at least two copies available of a program, you can do anything you want to your copy of that disk. If you do the worst, what damage have you done? Absolutely none since you can then simply reinsert another copy of the program or file and continue.
9. Not relaxing and enjoying. Become comfortable and relaxed with your machine. There is little you can do to the hardware, and if you back up your software, you can do no damage. So sit back and relax and enjoy.
10. Going it alone and not joining a users' group. There is so much to learn that it is hard for one person to teach herself. Don't be stubborn, ask for help. Even the most skilled computer hacker was once a novice. The users' group is a valuable resource for support and encouragement.
11. Not saving routinely and periodically. Murphy's Law holds true. What can go wrong does go wrong. If the power goes out, you turn off the computer, or accidentally erase a file, that data you so painstakingly typed into the computer is lost unless you saved the file to disk. And if it's important enough, save the work on two different disks. So SAVE, SAVE, SAVE.

This article has been adapted from many similar articles from many sources.

Reprinted from Upstate Apple Users

MEMBER MUG NEWS SERVICE ™

BACK 4 UP

ESSENTIAL DATA DUPLICATOR

Back up your copy-protected disks with **ESSENTIAL Data Duplicator 4 PLUS** ■ **EDD 4 PLUS** is new technology, not just 'another' copy program. The **EDD 4 PLUS** program uses a specially designed hardware card which works with your disk drives to back up disks by accurately copying the bits of data from each track. Don't be fooled . . . no other copy-program/system for Apples can do this! ■ In addition to backing up disks, **EDD 4 PLUS includes several useful** utilities such as examining disk drives, certifying disks, displaying drive speed rpm's, plus more! ■ **EDD 4 PLUS** runs on Apple II, II Plus (including most compatibles), and IIe, and is priced at $190.00 (duodisk/unidisk 5.25 owners must add $32.00 for a special cable adapter). ■ A standard **EDD 4** version which doesn't include any hardware is available, and can be used on Apple IIc and III (using emulations mode) and is priced at $125.00.

■ Bankcard and Master card accepted by phone. Add $3.00 to all orders for postage and handling.

All orders must be prepaid.

WARNING: EDD is sold for making archival copies only

UTILICO SOFTWARE

83 HALL ST., BONDI BEACH,
NSW 2026 PHONE (02) 30 2105